USER TEMPLATE MODE FOR INTERACTIVE GRAPHICS

U.G. Lama
Department of Fisheries and Environment

Abstract

The continuing spread of low-cost interactive graphics from scientific and engineering fields into less technical areas has increased the demand for application and user-oriented software that requires little or no knowledge of programming. Thus, the conundrum presented to the systems designer is how to tailor a module to a particular application and yet at the same time make the system flexible enough to be applicable to different fields.

The solution which we have used effectively in our shop, can be called "User Template Mode". In this system from a set of generalized building blocks, by means of a simple catechism, a user can initially select all the desired parameters and options. Thereafter, he need only invoke the template to automatically obtain his "customized" interactive display. This concept has been applied to a diverse range of problems, from program forecasting to log-sawing simulation.

LA PRODUCTION DE GRAPHES À L'AIDE D'UN PATRON DYNAMIQUE

Résumé

L'apparition continuelle de système peu coûteux (système permettant la réalisation de graphiques dynamiquement) provenant des domaines scientifiques et de l'ingénierie dans des domaines moins techniques a augmenté la demande de logiciel appliqué et orienté vers l'usager; ce logiciel demande peu ou pas de connaisance en programmation. Le problème présenté aux analystes de système fût de construire un module pour une application particulière mais que ce module soit assez souple pour qu'on puisse l'utiliser dans plusieurs domaines d'application.

La solution utilisée dans notre section peut être appelé «Un patron dynamique». A partir d'une série de modules généraux, par la connaissance d'un langage de contrôle simple, l'usager choisit premièrement tous les paramètres et les options dont il aura besoin. Après, il n'a qu'à appeler le modèle désiré pour obtenir automatiquement sa représentation graphique. Ce concept a été appliqué à différents niveaux d'application, allant des programmes de prédiction jusqu'à la simulation du sciage des bûches.

USER TEMPLATE MODE FOR INTERACTIVE GRAPHICS

INTRODUCTION

With the advent of low-cost interactive graphics, the application of computer graphics has spread from scientific and engineering fields into less technical areas. Though for many of these problems a pictorial means of communication is far more natural than alphanumeric, it introduces a number of special problems. Formally, the user usually knew enough FORTRAN to write "Calcomp-like" routines, or some graphics language, such as IMAGE, GINO-F, or GRAPPLE, in order to produce the desired plots either on an actual plotter or a screen. However, there is a rapidly growing segment of users and potential users, who have no knowledge of computing. Thus, the problem presented to the interactive graphics-systems designer is how to "spoon-feed" a particular application and yet at the same time make the system flexible enough to be applicable to different fields--a seemingly impossible task.

The solution which we have used effectively in our shop, can be called "User Template Mode." In this system the interactive graphics software is set up in a very general way under control of the user via a simple program catechism. In this mode the user has a very wide range of options; however, the user with a specific application does not usually want to decipher or concern himself with a myriad of possibilities, he wants a quick solution to his problem. To achieve this end, he can simply choose one of the common template files (such as three views in 3-D) from the program library or use the default settings. However, should none of these satisfy his special requirement, he or one of his staff can run through the program once with the desired settings and options. Thereafter, he need only invoke the template mode and all these steps become automatic.

Though not patterned after the "firmware" concept in computer design, it shares many of its features, in that it allows one to form a generalized set of building blocks whose total function need be designated only when applied to a specific problem. In hardware terms one could describe it as "burning-in" a particular logic pattern; however, in this case it is a "soft burn".

THE PLOT3D MODULE

The use of the template mode can be best illustrated by explaining its implementation in a specific package, such as PLOT3D. PLOT3D is a general purpose 3D plotting package that enables non-programmers to easily display their data or program results. This module is set up in such a way that the image can be manipulated with respect to the angle of view, position, size, scale, inversion, number, picture style, etc., from the keyboard by means of simple English (could be French) phrases.

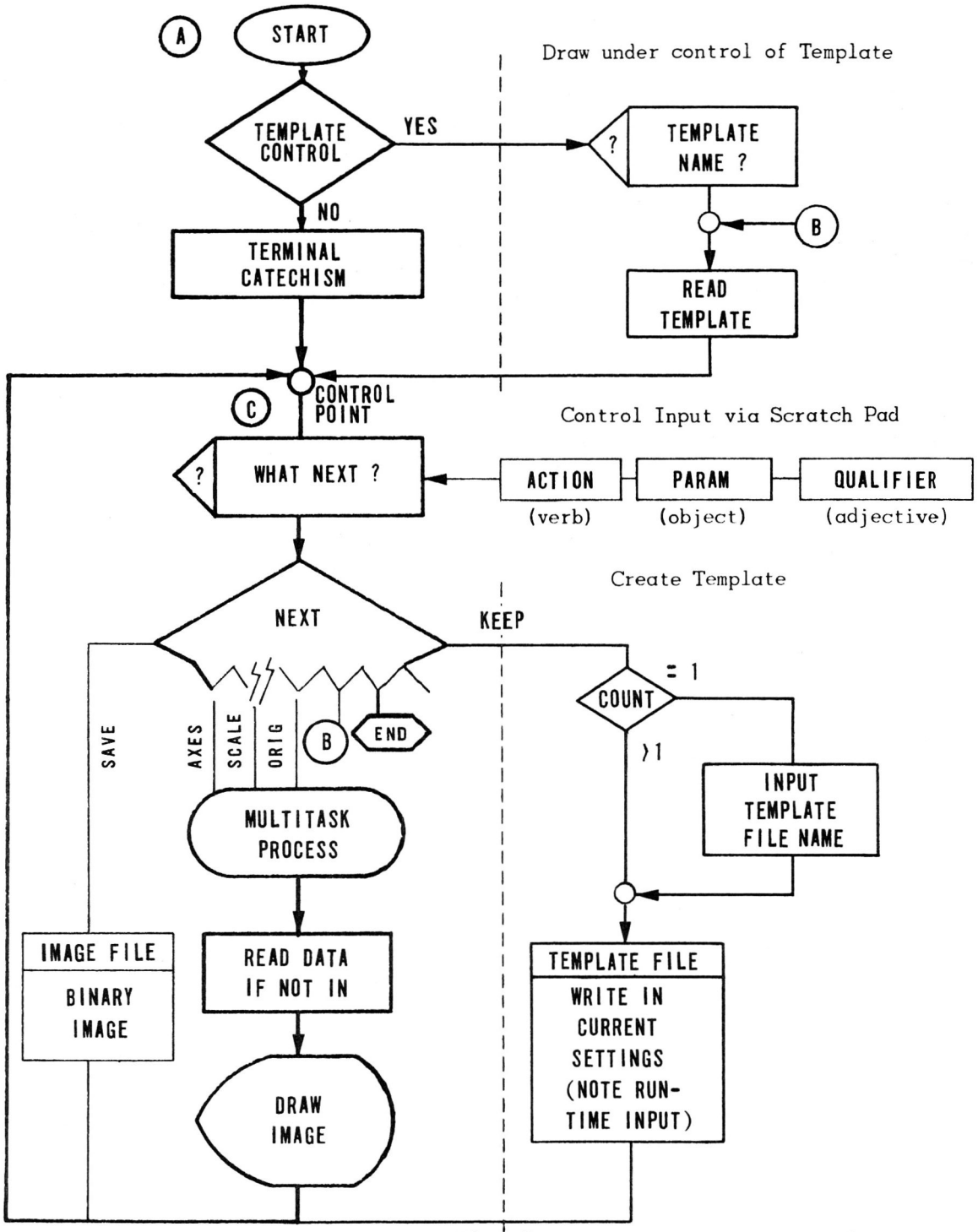

Fig. 1

Figure 1 illustrates the overall control logic of PLOT3D and how the template subsystem is interfaced to it. The principal program flow is as follows. On entering the program the user has a choice of setting the various parameters from the keyboard or entering them from a template file. After reading-in the data-file, computing the necessary transforms, and displaying the image on the screen, the cursor returns to the scratch pad indicating that the program has returned to the main control point C.

The Command String

Program control and image manipulation is directed mainly through this principal control node, at which the user is prompted with "WHAT NEXT?" in the scratch pad. In response to this query the program expects a one to four word English phrase, such as CHANGE SIZE or OVERLAY NEW ORIGIN. The structure of this command string is:

ACTION on PARAMETER subject to QUALIFIERS

The command is composed by choosing one or more words from the following:

Word 1	Word 2	Word 3
HELP	FILE	
END	PICSTYLE	1
SAVE	TITLE	2
DELETE	MAXMIN	
KEEP	LENGTH	
NEW	SIZE	
CHANGE	ORIGIN	
INVERT	ANGLE	
AXES	XY	
BOX	YZ	
GRID	ZX	
FRAME		

If any of these commands is preceded by OVERLAY, the screen will not be erased, so that any number of multiple images can be created.

The advantage of this structure is that it is easy to use and remember because of the naturalness of the words. Secondly, in the interest of speed and convenience only the initial character of each word is essential. Furthermore, this command structure in conjunction with the central control-point concept is inherently open-ended, allowing for easy inclusion of additional commands and modifiers.

Without delving into the niceties of what each combination of words produces, suffice it to say that any or all of the parameters of word 2 can be changed to produce another image on the screen alone or together with the previous ones. Axes, grids, and boxes can be superimposed. Images can be inverted, right-hand axial systems can be transformed to left-hand axial systems, and data matricies can be transposed with respect to X and Y, and so on.

Creating a Template or Command File

When a user wishes to "keep" any of the current plotting parameters for future application, he simply responds "KEEP" to the "WHAT NEXT?" question before issuing another manipulation command, if any, as indicated in the flow chart. In this way a user can experiment with and build up a whole series of images with different orientations and origins, but commit to template only the settings for a select group of views. To provide the necessary link between writing the template and reading it, the user is given the opportunity to assign a meaningful name to his template on initially asking for "KEEP". The template created in this way can now be accessed, as will be shown in the applications section, to generate similar layouts for as many data sets as the user wishes to display.

The usefulness of the template has been enhanced by allowing one of the parameters to remain as a variable, to be entered at plot time. For instance, "KEEP MAX" would allow one to set up a multiple-image plot, whose scaling maxima could be altered, at run time, to suit different sets of data.

One could easily incorporate additional features into the template, such as specific labels or even the terminology for the program catechism, tailored to a particular application. However, one must guard against unnecessary complications as the increased overhead and creation effort may defeat the intended purpose - simplicity. As it stands, there is very little processing "overhead", since this method does not monitor all the actions of the user but rather records a select group of parameters at the end of this activity.

APPLICATIONS

The areas to which we have applied this User Template concept range from departmental program forecasting to log-sawing simulation. In the former the users are planners and financial specialists, who speak in terms of activity structures and functional correlations; whereas, in the latter case, the users are statisticians and forestry scientists. Hence, success depends rather crucially on the ease of use and universality of the dialogue.

a) Display of Data Files

The large degree of simplification introduced by this method and the nature of the PLOT3D module is illustrated by the following sample run. To display, for a number of Services in the Department, a surface of expenditure against years for several activities, a planner need only answer a few simple questions as follows: (answers are underlined)

-PLOT3D.

** 3D ARRAY PLOTTER ** vers x.x.xx

COMMAND SOURCE - TERM (CR) or FILE? F
TEMPLATE FILENAME: VU3
DATA FILENAME: SER
Title: SER DOLLARS BY FUNCTION

At this point the results are plotted as shown below.

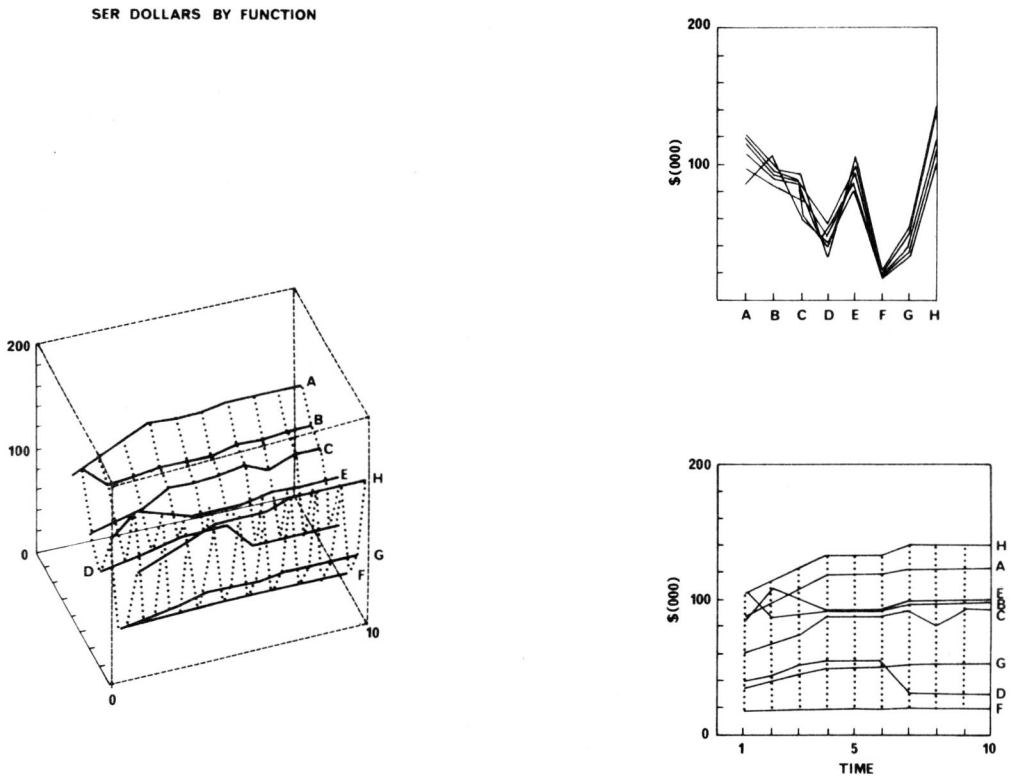

Fig. 2

Subsequently, in the scratch pad he would have the opportunity to change any of the parameters, superimpose other data for comparison, extract a particular curve for closer scrutiny, or to plot other data in the same layout by entering the new data filename as follows:

SAME TEMPLATE - Another File? Y
DATA FILENAME: AES

In this way the planner of forecaster can "browse" through the data sets almost like the pages of a book. Implicit in this ability to compare and examine different data sets is an adherance to some "universal" standard of data-file organization.

Traditionally, this data is published as a thick volume of tables, which are difficult to digest, especially in terms of trends, relative significance and totality. On the other hand, the graphical method is ideal for summarizing a large amount of related data, as it emphasizes trends, obviates distraction by unnecessary details and highlights anomalies and errors. For instance, if one examines figure 2, one can see that the expenditure for all activities is simply a linear extrapolation beyond year 7; however, activity C has a kink in it. Re-examination of the data showed that this was due to the digit 8 being entered instead of a 9 in the relavent number.

b) Display of Functions

As researchers are commonly trying to visualize the behavior of complex mathematical functions, a simple method of entering formulae is provided. By choosing one of the more common templates from the library, the user can quickly get an overview of his function, which allows him to easily zero-in on the region of most interest. As a further refinement he is able to vary the resolution of the net draping the function. For instance, one may wish to investigate the interaction of the two terms in the following function, as the value of d is varied.

$$z = a \cos^2 (x-d) + b \cos^2 (x+d) \cdot \cos^2 y$$

For d near $\Pi/2$, this function is a "sea of uniform rounded peaks" in the xy-plane, so that only a couple of periods need be examined, as displayed in figure 3.

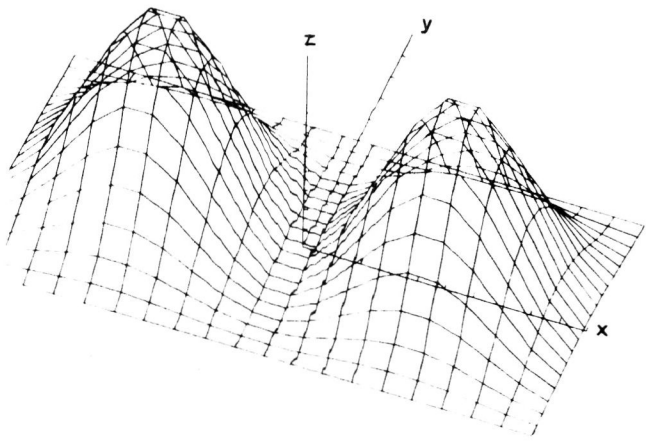

Fig. 3 Region x $\pm\Pi$, y $\pm\Pi/2$, z = 0 to 1

Because of the flexibility of multiple-imaging, the user can even display what is strictly speaking a 4-dimensional function. For instance, figure 4 shows the bahavior of the following formula, as the fourth-dimension w is assigned values 1, 5, 10, and 20.

$$z = 100. \left\{ \{w - \frac{10}{2\Pi} \tan^{-1} (y/x)\}^2 + \{(x^2+y^2)^{\frac{1}{2}}-1\}^2 \right\} + w^2$$

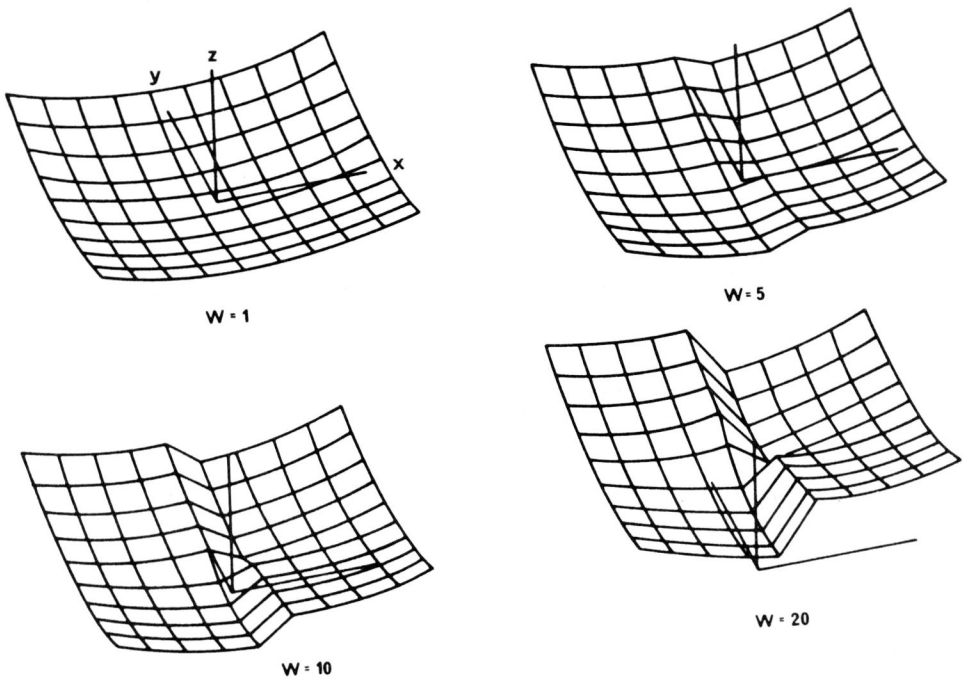

Fig. 4 End values for axes: x = y = 10, z = 50000

Even in the case of functions, no knowledge of programming is required apart from being able to express an algebraic formula in FORTRAN annotation.

CONCLUSIONS

In designing such an interactive system the major effort has to be devoted to the communications aspect, both internally and between man and machine, rather than in the graphics component. The system has to be "forgiving", so that the user has correction opportunities at many levels. Sufficient prompting help has to be inserted in the program, for users do not read manuals unless absolutely necessary. On the other hand, the number of prompts has to be minimized for the experienced user and for production runs. A convention for data files has to be adopted, so that information can be accessed across packages and accounts. At the same time, the program has to be sufficiently flexible to accept several file types and sufficiently "intelligent" that the user does not require "a priori" knowledge of his data as to its number or magnitude. In short, the ease of use of such a compre-

hensive package is usually inversely proportional to its internal complexity.

As for the template itself, the exact format does not appear to be crucial; however, it would be advantageous that it have the following properties:

I-from the users' point of view,
- -be easy to apply - invoke simply by name or key
- -be easy to create, NOT requiring any knowledge of its internal structure or programming
- -have a degree of flexibility
- -cater to the users' needs rather than the systems analyst's preconceived notions.

II-from the systems point of view,
- -require a minimum of overhead
- -be printable (i.e. less than 132 characters per record)
- -be readable (i.e. alphanumeric, NOT binary or hexadecimal)
- -be easily modifiable and chainable via the normal system editor
- -be open-ended with respect to the addition of future parameters, preferably without invalidating older versions.

The extra simplification brought to interactive graphics by the template mode has introduced the power of the graphics method to several groups that formerly would not "touch" a terminal. Once the barrier is breached it is amazing what level of sophistication the neophyte attains in just a few sessions.